

Unit Tests

Table of Contents

| | |
|---|---|
| 1. Ming64 & MSYS2 | 1 |
| 1.1. Installation | 1 |
| 2. Fast set unit tests | 1 |
| 2.1. Use case: one class with a couple of methods to be tested | 1 |
| 3. Google Mock | 3 |
| + = Google Test and Google Mock Doc Writer < christian.popescu@outlook.com > | |
| v 1.0, 2019-12-14 | |

1. Ming64 & MSYS2

1.1. Installation

Install package mingw-w64-x86_64-gtest

```
pacman -S mingw-w64-x86_64-gtest
```

[Package mingw-w64-x86_64-gtest](#)

2. Fast set unit tests

2.1. Use case: one class with a couple of methods to be tested

1. Install Google Test :)
2. Create a simple class to be tested

```

#ifndef CPP_LIB_STRINGBUILDER_H
#define CPP_LIB_STRINGBUILDER_H

#include <string>

namespace cppstdlib {
    class StringBuilder {
    public:
        StringBuilder();
        virtual ~StringBuilder();
        std::string getString(); // returns StringBuilder content as string
        int getLength(); //
        int getSize(); // get the allocated size
        void append(std::string str); // append of string

    private :
        char* theString;
        int size;
        int length;
        const static int minAlloc;
    };
}

#endif // CPP_LIB_STRINGBUILDER_H

```

3. Create a simple test file that will keep the tests

```

#include "gtest/gtest.h" // include GoogleTest
#include "StringBuilder.h" // include class to test

namespace {
    // Test default constructor
    TEST(stringBuilder, DefaultConstructor) {
        cppstdlib::StringBuilder sb;

        EXPECT_EQ(0, sb.getLength());

        EXPECT_EQ(16, sb.getSize());
    }
}

```

+ 4. Build and run the test package (example with makefile)

```

GTEST_CFLAGS = `pkg-config --cflags gtest_main`
GTEST_LIBS = `pkg-config --libs gtest_main`

.PHONY: tests all

tests: all
    ./StringBuilder_UnitTests

all: StringBuilder_UnitTests

StringBuilder_UnitTests: StringBuilder_UnitTests.o StringBuilder.o
    $(CXX) $(CXXFLAGS) $(LDFLAGS) $^ -o $@ $(GTEST_LIBS)

StringBuilder_UnitTests.o: StringBuilder_UnitTests.cc
    $(CXX) $(CPPFLAGS) $(CXXFLAGS) $< -c -o $@ $(GTEST_CFLAGS)

StringBuilder.o: StringBuilder.cc
    $(CXX) $(CPPFLAGS) $(CXXFLAGS) $< -c -o $@

```

+ 5. Run example

```

$ make -f makefile_test
g++  StringBuilder_UnitTests.cc -c -o StringBuilder_UnitTests.o `pkg-config --cflags gtest_main`
g++  StringBuilder_UnitTests.o StringBuilder.o -o StringBuilder_UnitTests `pkg-config --libs gtest_main`
./StringBuilder_UnitTests
Running main() from C:/repo/mingw-w64-gtest/src/googletest-release-1.8.1/googletest/src/gtest_main.cc
[=====] Running 1 test from 1 test case.
[-----] Global test environment set-up.
[-----] 1 test from stringBuilder
[ RUN      ] stringBuilder.DefaultConstructor
[      OK  ] stringBuilder.DefaultConstructor (0 ms)
[-----] 1 test from stringBuilder (0 ms total)

[-----] Global test environment tear-down
[=====] 1 test from 1 test case ran. (0 ms total)
[  PASSED  ] 1 test.

```

3. Google Mock

Examples :

[Docs on Git Hub Google Mock](#)